NORTH-HOLLAND

A Model for CAGD Using Fuzzy Logic*

J. Jacas and A. Monreal
Sec. Matemàtiques i Informàtica, E.T.S. Arquitectura de Barcelona, Barcelona, Spain

J. Recasens
Sec. Matemàtiques i Informàtica, E.T.S. Arquitectura del Vallès, Sant Cugat, Spain

ABSTRACT

This paper presents a first approach to a system based on fuzzy logic for the design of curves and surfaces in the context of computer aided geometric design. Bézier curves and surfaces can be seen as particular cases of this system.

KEYWORDS: ***computer aided geometric design, Bézier curves, Bézier surfaces, approximate reasoning, fuzzy members, t-norms, fuzzy control***

1. INTRODUCTION

In computer aided geometric design (CAGD), usually the user provides a set of points and the program produces a curve or surface whose shape is controlled by these points. In Figures 1 and 2, we can see two types of desired results: with the same set of points, in Figure 1 a curve interpolating the points is obtained, while in Figure 2 the curve is only controlled in some way by the points.

Since this is a design process, it is essential to control the degree of smoothness of the solution.

* Research partially supported by the DGICYT project PB94-1208.

Address correspondence to Professor Joan Jacas, Secció Matemàmatiques i Informàtica, E.T.S. Arquitectura de Barcelona, Avda. Diagonal 649, 08028 Barcelona, Spain. E-mail: `jacas@ea.upc.es`.

Received March 1, 1996; accepted October 1, 1996.

International Journal of Approximate Reasoning 1997; 16:289–308

655 Avenue of the Americas, New York, NY 10010

0888-613X/97/$17.00
PII S0888-613X(96)00124-7

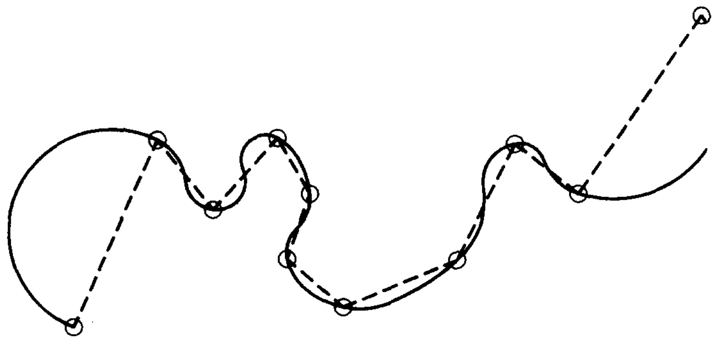


Figure 1.

In this paper, we will pay attention to the following four problems:

PROBLEM 1a (Nonparametric or functional curves) *Given* $n + 1$ *points of control* $\vec{P_i} = (a_i, b_i) \in \mathbb{R}^2$, $i = 0, 1, \ldots, n$, *with* $a_0 < a_1 < \cdots < a_n$, *we want to construct the graph of a function* $f : [a_0, a_n] \to \mathbb{R}$ *such that* $\{a_i\}$ *are values of the independent variable and* $\{b_i\}$ *are values or approximations of* $\{f(a_i)\}$.

PROBLEM 1b (Parametric curves) *Given* $n + 1$ *points of control* $\vec{P_i} = (a_i, b_i) \in \mathbb{R}^2$ *or* $\vec{P_i} = (a_i, b_i, c_i) \in \mathbb{R}^3$, $i = 0, 1, \ldots, n$, *we want to construct* $n + 1$ *values to* $t_0 < t_1 < \cdots < t_n$ *of a parameter and a parametric curve* $\vec{f} : [t_0, t_n] \to \mathbb{R}^2$ *or* $\mathbb{R}^3$ *in such a way that the points* $\{\vec{P_i}\}$ *are the values or approximations of* $\{\vec{f}(t_i)\}$.

PROBLEM 2a (Nonparametric or functional surfaces) *Given* $n + 1$ arbitrarily distributed *points of control* $\vec{P_i} = (a_i, b_i, c_i) \in \mathbb{R}^3$, $i = 0, 1, \ldots, n$, *we want to construct a surface passing through or controlled by these points*

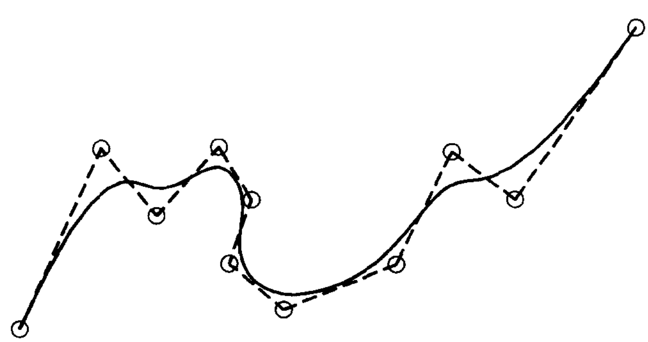


Figure 2.

as the graph of a map $f: A \subset \mathbb{R}^2 \to \mathbb{R}$ *(for example,* $A = [a_{\min}, a_{\max}] \times [b_{\min}, b_{\max}]$*) such that* $\{(a_i, b_i)\}$ *are the values of the independent variables and* $\{c_i\}$ *values or approximations of* $\{f(a_i, b_i)\}$.

PROBLEM 2b (Parametric surfaces) *Given a net of* $(n + 1) \times (m + 1)$ *points* $\vec{P}_{ij} = (a_{ij}, b_{ij}, c_{ij}) \in \mathbb{R}^3$, $i = 0, \ldots, n$, $j = 0, \ldots, m$, *we want to construct* $n + 1$ *values* $s_0 < s_1 < \cdots < s_n$ *of a parameter* s *and* $m + 1$ *values* $t_0 < t_1 < \cdots < t_m$ *of a parameter* t *and a parametric surface* $\vec{f}: [s_0, s_n] \times [t_0, t_m] \to \mathbb{R}^3$ *in such a way that the points* $\{\vec{P}_{ij}\}$ *are the values or approximations of* $\{\vec{f}(s_i, t_j)\}$.

There are many standard techniques to treat the preceding problems. For more complete information, also about methods to construct values of the parameter in parametric cases, the reader is referred to [3–5, 8].

The common algorithm of all design methods that interact between the procedure and the user is shown in Figure 3. The algorithm shows the existing vagueness of the data points and its relation with the goal curves. Therefore, it is natural to treat the problem in the setting of approximate reasoning. The model proposed in Section 2 uses fuzzy numbers and Takagi-Sugeno techniques in fuzzy control to design curves and surfaces. Some examples are shown in Section 3, and in Section 4 it is proved that Bézier curves and surfaces can be thought of as particular cases of the model.

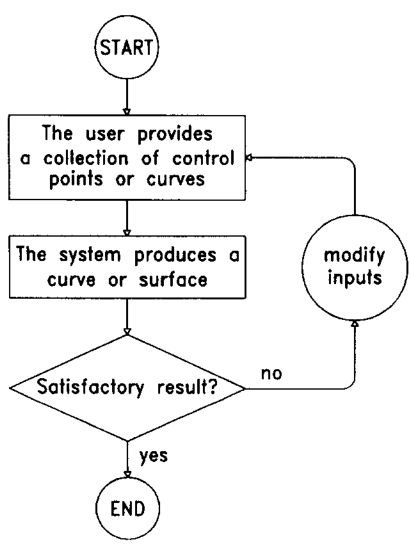


Figure 3.

2. THE MODEL

We propose a system based on fuzzy linguistic rules used in fuzzy control [2, 11, 7]. We have adopted the point of view of Takagi and Sugeno [7, 10] in order to avoid the defuzzification process.

The main idea is to use rules of the following type:

If $\vec{x}$ is close to $\vec{x}_i$ in the domain, then $\vec{y}$ is close to $\vec{y}_i$ in the image.

This idea is developed in the following model:

- Use $n + 1$ rules

$$R_i : \text{if } \vec{x} \simeq \vec{x}_i \text{ then } \vec{y} \simeq \vec{y}_i, \qquad i = 0, 1, \ldots, n.$$

- Model the degree of fulfilment of the antecedent of the ith rule with a fuzzy set α_i such that $\alpha_i(\vec{x}_i) = 1$ and $\alpha_i(\vec{x})$ is a decreasing function of distance to $\vec{x}_i$ [6].
- The consequence function of each rule, $f_i(\vec{x})$, is the constant $\vec{y}_i$. (Therefore, the system coincides with the *height method*).
- The output of the system is obtained as a mean of the outputs of each rule weighted by the fuzzy sets α_i:

$$\vec{y}(\vec{x}) = \frac{\sum_{i=0}^{n} \alpha_i(\vec{x}) \cdot \vec{y}_i}{\sum_{i=0}^{n} \alpha_i(\vec{x})} = \sum_{i=0}^{n} F_i(\vec{x}) \cdot \vec{y}_i \tag{2.1}$$

where $F_i(\vec{x}_i) = \alpha_i(\vec{x}) / \sum_{j=1}^{n} \alpha_j(\vec{x})$.

So the result is a convex linear combination of the outputs of the rules where the coefficients depend on the data points and the fuzzy sets α_i.

2.1. Remarks

2.1.1 If the domain is in $\mathbb{R}$ (the case of a curve), then $\alpha_i = \mu_i$ are fuzzy numbers.

2.1.2 If the domain is in $\mathbb{R}^2$ (the case of a surface), then the closeness α_i of a point $\vec{x} = (x, y)$ to $\vec{x}_i = (x_i, y_i)$ can be modeled in the following way: $\vec{x} = (x, y)$ is close to $\vec{x}_i = (x_i, y_i)$ if and only if x is close to x_i and y is close to y_i. That is, if μ_i and ν_i are fuzzy numbers modeling the closeness of x and y to x_i and y_i respectively, then the fuzzy set α_i can be defined as $\alpha_i(\vec{x}) = T(\mu_i(x), \mu_i(y))$, where T is a suitable t-norm that plays the semantic role of a conjunction. In order to maintain the degree of smoothness of the resulting surface inherited from the smoothness of the fuzzy numbers, it is convenient to choose a C^∞ t-norm. In our model, we use the *product*.

2.1.3 The fuzzy sets $\{\mu_i\}$ must cover the domain in the sense that for every point $\vec{x}$ in the domain, there must exist a fuzzy set μ_i with $\mu_i(\vec{x}) \neq 0$. This is a technical requirement in order to avoid dividing by zero in (2.1) and has the meaning that the image of every point of the domain must be controlled by at least one of the given points $\{\vec{x}_i\}$.

2.1.4 Curves of the types in Problems 1a and 1b generated by this model cannot be simultaneously interpolative and smooth (i.e. tangent continuous). Indeed, since the curve always lies in the convex hull $C(\{\vec{P}_i\})$ of the control points $\vec{P}_i$, the interpolative solution will necessarily present a cusp in some point $\vec{P}_k$.

2.1.5 It is worth noticing that, whereas in the classical methods used in CAGD only the selection of points is important, here we also have the possibility of choosing the type and size of the fuzzy number associated to each point, even within the same set of nodes. This aspect gives great versatility to our model.

With all this in mind, the solutions proposed to the problems of the introduction are the following:

2.2. Solutions

PROBLEM 1a (nonparametric curves). In this case, $\vec{P}_i = (a_i, b_i)$, the domain is some interval in $\mathbb{R}$, $\vec{x}_i = a_i$, $\vec{y}_i = b_i$ and $\alpha_i = \mu_i$ are normalized fuzzy numbers. The solution curve is

$$y = f(x) = \frac{\sum_{i=0}^{n} \mu_i(x) \cdot b_i}{\sum_{i=0}^{n} \mu_i(x)} = \sum_{i=0}^{n} F_i(x) \cdot b_i. \tag{2.2}$$

PROBLEM 1b (parametric curves). In this case, $\vec{P}_i = (a_i, b_i) \in \mathbb{R}^2$ or $\vec{P}_i = (a_i, b_i, c_i) \in \mathbb{R}^3$. The domain is an interval in $\mathbb{R}$, the variable is a parameter t, $\vec{x}_i = t_i$, $\vec{y}_i = \vec{P}_i$, and $\alpha_i = \mu_i$ are normalized fuzzy numbers. The solution curve is

$$\vec{y} = \vec{f}(t) = \frac{\sum_{i=0}^{n} \mu_i(t) \cdot \vec{P}_i}{\sum_{i=0}^{n} \mu_i(t)} = \sum_{i=0}^{n} F_i(t) \cdot \vec{P}_i. \tag{2.3}$$

PROBLEM 2a (nonparametric surfaces). In this case, $\vec{P}_i = (a_i, b_i, c_i) \in \mathbb{R}^3$, the domain is some region on $\mathbb{R}^2$, $\vec{x}_i = (a_i, b_i)$, $\vec{y}_i = c_i$, and $\alpha_i(x, y) = \mu_i(x)\nu_i(y)$, where μ_i, ν_i are normalized fuzzy numbers. The solution surface is

$$z = f(x, y) = \frac{\sum_{i=0}^{n} \alpha_i(x, y) \cdot c_i}{\sum_{i=0}^{n} \alpha_i(x, y)} = \sum_{i=0}^{n} F_i(x, y) \cdot c_i. \tag{2.4}$$

PROBLEM 2b (parametric surfaces). In this case, $\vec{P}_{ij} = (a_{ij}, b_{ij}, c_{ij})$; the domain is a rectangle in $\mathbb{R}^2$; the variables are the parameters $s \in [s_0, s_n]$, $t \in [t_0, t_m]$; and we have $\vec{x}_{ij} = (s_i, t_j)$, $\vec{y}_{ij} = \vec{P}_{ij}$, and $\alpha_{ij}(s,t) = \mu_i(s)\nu_j(t)$, where μ_i, ν_j are normalized fuzzy numbers. The solution surface is

$$\vec{P}(s,t) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} \alpha_{ij}(s,t) \cdot \vec{P}_{ij}}{\sum_{i=0}^{n}\sum_{j=0}^{m} \alpha_{ij}(s,t)} = \sum_{i=0}^{n} \sum_{j=0}^{m} F_{ij}(s,t)\vec{P}_{ij}, \qquad (2.5)$$

where

$$F_{ij}(s,t) = \frac{\alpha_{ij}(s,t)}{\sum_{i=0}^{n}\sum_{j=0}^{m} \alpha_{ij}(s,t)}.$$

Note: Since in this case the set of points determines a net, we have used a double index for convenience. If we want to use a single index k, we can, for instance, reassign indices in the following way:

$$\{i,j\} \to \{k\} \qquad \text{with} \quad k = i(m+1) + j.$$

3. EXAMPLES

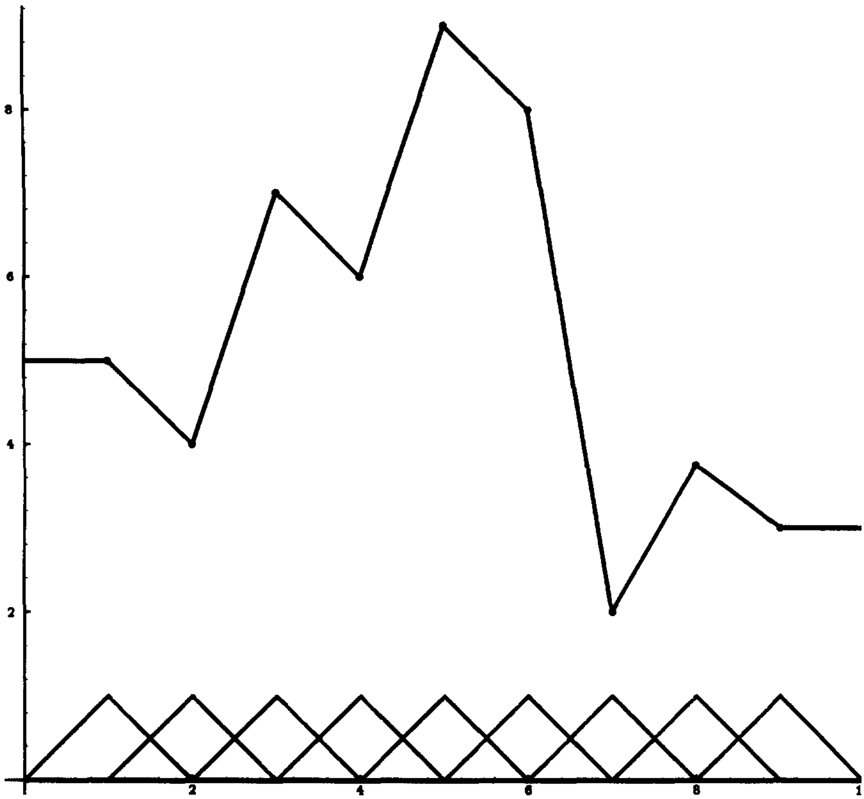
EXAMPLE 1 Given nine points of the plane, we construct an interpolating nonparametric curve (Figure 4). Since the fuzzy sets used are triangular (and therefore only of class C^0), the curve is polygonal (of class C^0 also).

EXAMPLE 2 With the same points of the preceding example but other fuzzy numbers representing the fuzzy points, we obtain a different curve, Figure 5 (also C^0, since the fuzzy numbers are also triangular).

EXAMPLE 3 Using Gaussian fuzzy numbers [$\mu_i(x) = e^{-[k_i(x-a_i)]^2}$] which are of class C^∞, we obtain a C^∞ nonparametric curve controlled by 10 points (Figure 6).

EXAMPLE 4 Using two different sets of Gaussian fuzzy numbers, we obtain two different parametric curves controlled by a sequence of 10 points (Figure 7).

EXAMPLE 5 Figure 8 shows two views of a surface controlled by eight points. Since, in this example, μ_i and ν_i are Gaussian fuzzy numbers, the surface is C^∞.



$$\mu_0(x) = \begin{cases} 1 & \text{if } x \le a_0 \\ \dfrac{a_1 - x}{a_1 - a_0} & \text{if } a_0 < x < a_1 \\ 0 & \text{if } a_1 \le x \end{cases}$$

$$\mu_n(x) = \begin{cases} 0 & \text{if } x \le a_{n-1} \\ \dfrac{x - a_{n-1}}{a_n - a_{n-1}} & \text{if } a_{n-1} < x < a_n \\ 1 & \text{if } a_n \le x \end{cases}$$

$$\mu_i(x) = \begin{cases} 0 & \text{if } x \le a_{i-1} \\ \dfrac{x - a_{i-1}}{a_i - a_{i-1}} & \text{if } a_{i-1} < x \le a_i \\ \dfrac{a_{i+1} - x}{a_{i+1} - a_i} & \text{if } a_i < x < a_{i+1} \\ 0 & \text{if } a_{i+1} \le x \end{cases} \qquad \sum_{i=0}^{n} \mu_i(x) \equiv 1 \quad \forall x$$

$$F_i(x) = \mu_i(x), \qquad i = 1, \ldots, n-1$$

Figure 4.

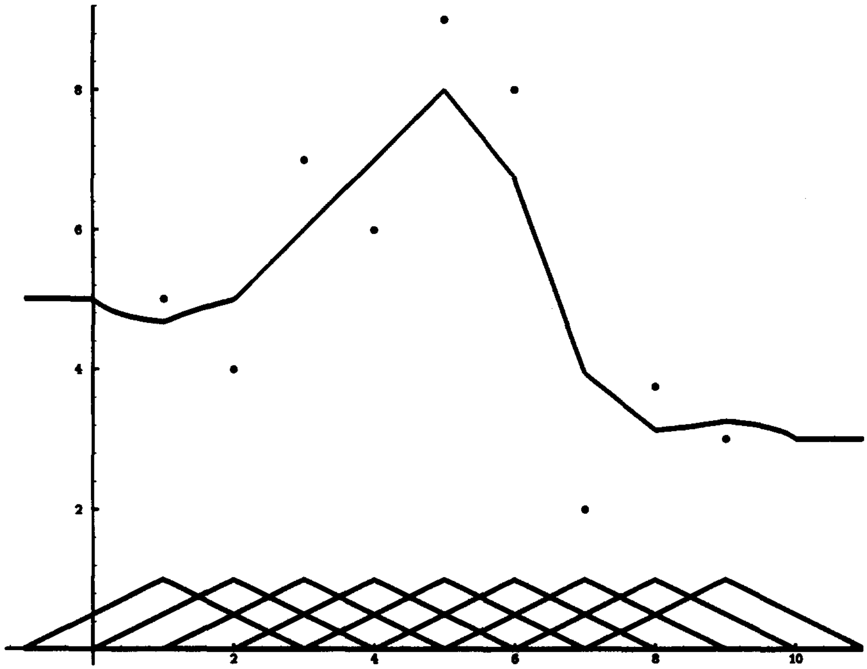
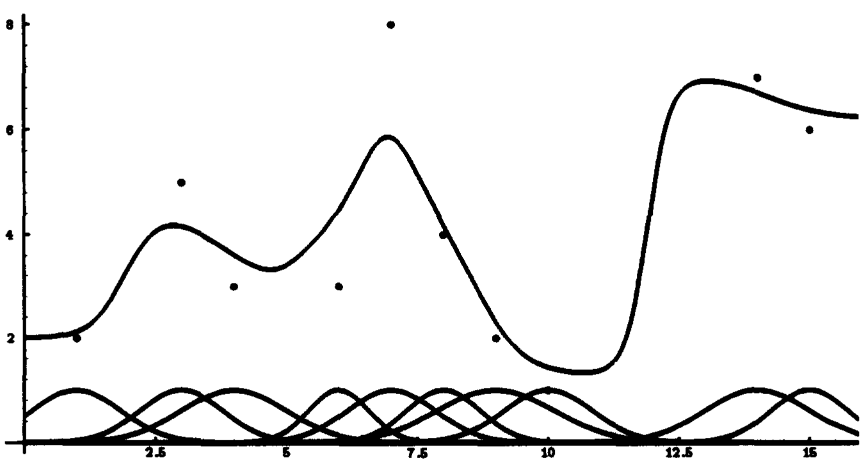
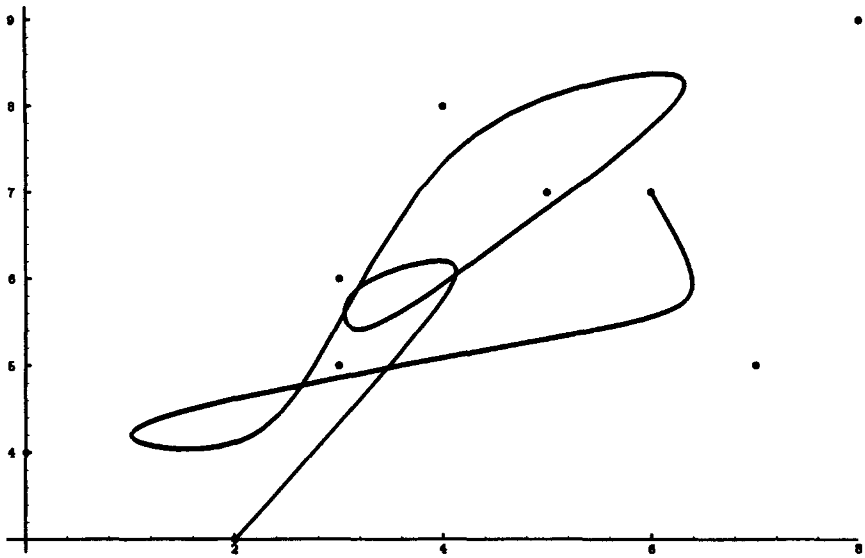


Figure 5.

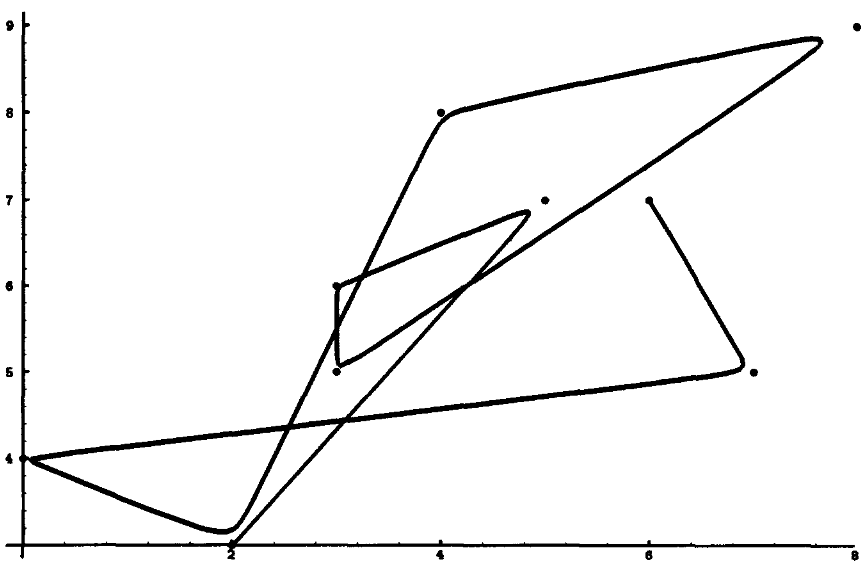


$$\mu_i(x) = e^{-[k_i(x-a_i)]^2}, \qquad i = 0, \ldots, n$$

Figure 6.



(a)



(b)

Figure 7.

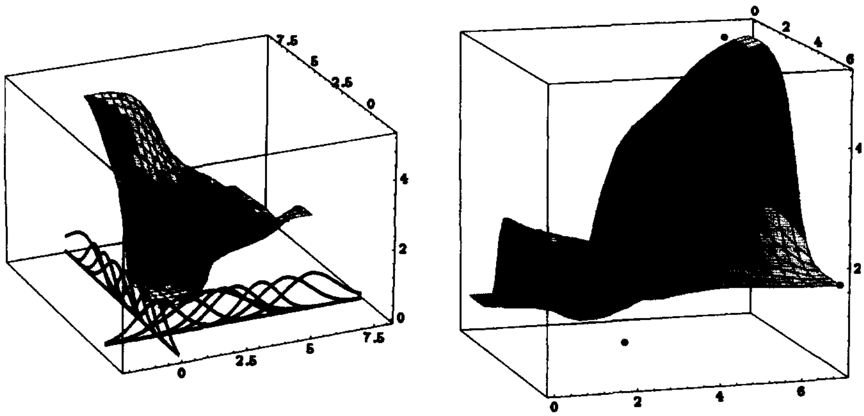


Figure 8.

It is worth noticing that in the functional case of Problem 2a, this model does not need to arrange the data points to form a grid; they can be situated arbitrarily. Also it is not necessary to triangulate the domain. These features make the model suitable, for example, for topographical problems.

4. BÉZIER CURVES AND SURFACES

In this section we will prove that Bézier curves and surfaces [3–5] can be seen as particular cases of our model.

We begin by focusing our attention on the case of curves. We translate the problem to be solved into a manageable form by means of Proposition 4.2; then we show the existence of solutions (Corollary 4.6) and give one of them within the proof of Theorem 4.10. The last part of this section simply takes advantage of the preceding work in order to solve the case of surfaces in Theorem 4.11.

LEMMA 4.1 *Let $\{\alpha_i\}$ be a family of fuzzy sets over the domain A fulfilling condition* 2.1.3 (*Section* 2.1), *and k a function from A onto $\mathbb{R}^+\backslash\{0\}$. Then the objects* (*curves or surfaces*) *generated using $\{\alpha_i\}$ and $\{\alpha_i/k\}$ coincide.*

Proof

$$\vec{y}(\vec{x}) = \frac{\sum_{i=0}^{n} \alpha_i(\vec{x})\vec{y}_i}{\sum_{i=0}^{n} \alpha_i(\vec{x})} = \frac{\sum_{i=0}^{n} \dfrac{\alpha_i(\vec{x})}{k(\vec{x})} \cdot \vec{y}_i}{\sum_{i=0}^{n} \dfrac{\alpha_i(\vec{x})}{k(\vec{x})}}. \qquad \blacksquare$$

In other words, if all the fuzzy sets are scaled in the same ratio at each point of A, the object obtained remains unchanged.

Let us recall that, given $n + 1$ control points $\vec{P_i}$, the Bézier curve $\vec{B}(t)$ controlled by them is obtained, in its Bernstein form, by means of the formula

$$\vec{B}(t) = \sum_{i=0}^{n} B_i^n(t) \cdot \vec{P_i} \qquad \text{for} \quad t \in [0,1], \tag{4.1}$$

where

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

is the ith Bernstein's polynomial of degree n (see [3]). To make (common) sense, we only consider the case $0 < n < +\infty$.

These curves are of the form (2.3) with $F_i = B_i^n$, without any reference to nodes $\{t_i\}$. If we wish to accommodate these curves into our model, we need to find appropriate normalized fuzzy numbers $\{\mu_i\}$ and nodes $\{t_i\}$ (the points where each μ_i reaches the value 1) such that

$$B_i^n = \frac{\mu_i}{\sum_{j=0}^{n} \mu_j}. \tag{4.2}$$

Figure 9 shows a typical Bézier curve with its control polygon together with the corresponding set of Bernsteins polynomials of degree 4.

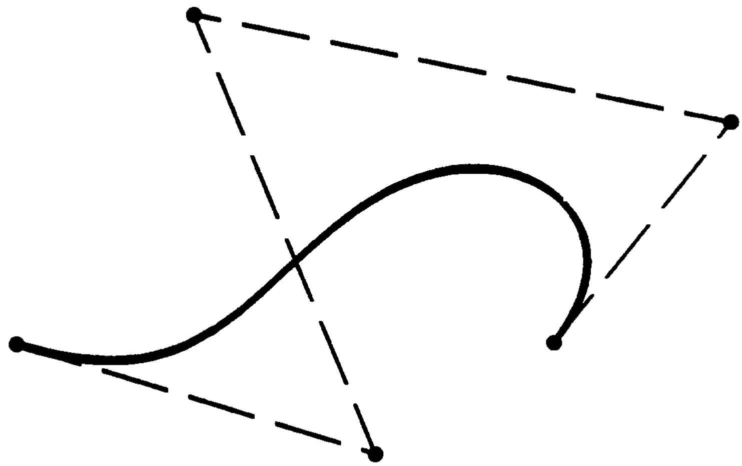
Since $\sum_{i=0}^{n} B_i^n(t) \equiv 1$, we could try to take $\mu_i = B_i^n$, but, in general, $B_i^n < 1$, i.e., $\{B_i^n\}$ cannot be seen as a family of normalized fuzzy numbers. Due to the semantics of our model, the antecedent $t \simeq t_i$ of each rule R_i must attain the truth value 1 for t_i. Fortunately, according to Lemma 4.1, we can use functions $k(t)$ in order to normalize the family $\{B_i^n\}$.

Therefore, our goal is to find a function $k : [0,1] \to R^+ \setminus \{0\}$ such that, $\mu_i = B_i^n / k$ is a fuzzy number and, for each i, there exists t_i such that $\mu_i(t_i) = 1$. As the order of the control points $\vec{P_i}$ must be kept, we need, in addition, the condition $0 \le t_0 < t_1 < \cdots < t_n \le 1$.

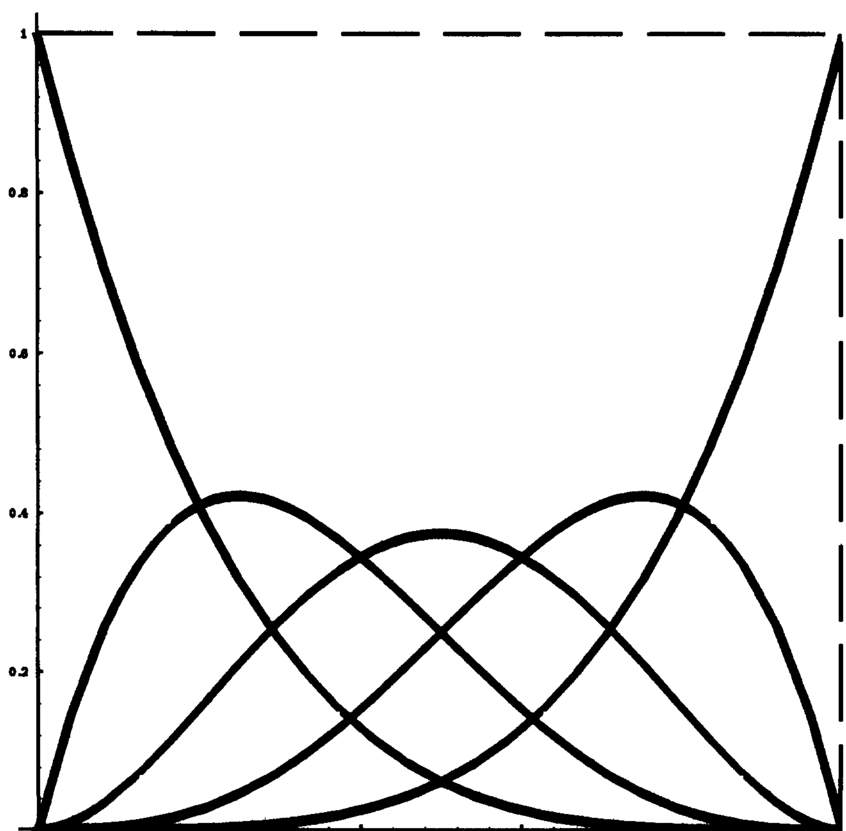
In particular, the preceding conditions for k mean that $k(t) \ge B_i^n(t)$ for every i and every t and that, for each i, there exists t_i verifying $k(t_i) = B_i^n(t_i)$ together with the condition $t_i < t_j$ if $i < j$, to ensure the order.

At this point, in order to prove Theorem 4.10, we only have to show that such functions k exist and give a way to construct one of them. The following proposition gives a necessary and sufficient condition for a function k to exist.

Proposition 4.2 *A function k, as required above, exists iff for each i there is a t_i such that if $j \ne i$ then $B_i^n(t_i) \ge B_j^n(t_i)$ and this set of $\{t_i\}$ can be chosen in such a way that $t_i < t_j$ when $i < j$.*



(a)



(b)

Figure 9.

Proof $\Rightarrow$: If for some i and every t we have $B_i^n(t) < B_j^n(t)$ for some j depending on t, then we cannot normalize the function $B_i^n(t)$ at any value t_i by dividing by $k(t_i)$, because then, for some j, $\mu_j(t_i) = B_j(t_i)/k(t_i) > 1$, contradicting the fact that μ_j is a fuzzy number.

If the $\{t_i\}$ exist but cannot be suitably ordered, the order of the points $\vec{P_i}$ cannot be preserved.

$\Leftarrow$: In this case, we can define $k_0(t) = \max B_i^n(t)$ (which is always greater than zero). Every function $k(t) \geq k_0(t)$ with $k(t_i) = k_0(t_i)$ accomplishes our requirements. This proves, in addition, that, if a solution exists, then it is not unique. ■

To prove the existence of solutions for k and to calculate one of them, we will embed the family $\{B_i^n(t)\}_{i \in \{0,1,\ldots,n\}}$ into the uniparametric family of functions $\{B_\lambda^n(t)\}_{\lambda \in [0,n]}$ where

$$B_\lambda^n(t) = \binom{n}{\lambda} t^\lambda (1-t)^{n-\lambda}, \qquad \text{with} \quad \lambda \text{ real},$$

$$\binom{n}{\lambda} = \frac{\Gamma(n+1)}{\Gamma(\lambda+1)\Gamma(n-\lambda+1)},$$

where Γ represents the gamma function [1].

Let us consider the parametric family of curves $y = B_\lambda^n(t)$, $\lambda \in [0, n]$, or $G_n(t, y, \lambda) = B_\lambda^n(t) - y = 0$ in its implicit form. For instance, Figure 10 displays the family $\{B_i^8(t)\}$. We can observe an upper envelope or outline over the polynomials which touches every one at a point. This outline will allow us to prove the existence of solutions and will be useful for the construction of one of them.

REMARK 4.3 In order to apply Proposition 4.2, since for $\lambda = 0$ and $t = 0$ we have $B_0^n(0) = 1 > 0 = B_\mu^n(0)$ if $\mu \in (0, n]$, we can take $t_0 = 0$. Likewise, since for $\lambda = n$ and $t = 1$ we have $B_n^n(1) = 1 > 0 = B_\mu^n(1)$ if $\mu \in [0, n)$, we can take $t_n = 1$. Thus, from now on, we can focus our attention on the case $\lambda \in (0, n)$ and $t \in (0, 1)$.

PROPOSITION 4.4 *For each* $\lambda \in (0, n)$ *there exists a* $t_\lambda \in (0, 1)$ *such that, for all* $\lambda' \in (0, n) \setminus \{\lambda\}$,

$$B_\lambda^n(t_\lambda) > B_{\lambda'}^n(t_\lambda) \quad \text{and} \quad \text{if } \lambda < \lambda' \text{ then } t_\lambda < t_{\lambda'}.$$

Proof Thinking of $B_\lambda^n(t)$ as a function of two variables (Figure 11), it is sufficient to show that for every λ, $B_\lambda^n(t)$ has a relative maximum, with respect to λ for some value t_λ, which is an absolute maximum too. This is related to the existence of the envelope of the family G_n (see [4]).

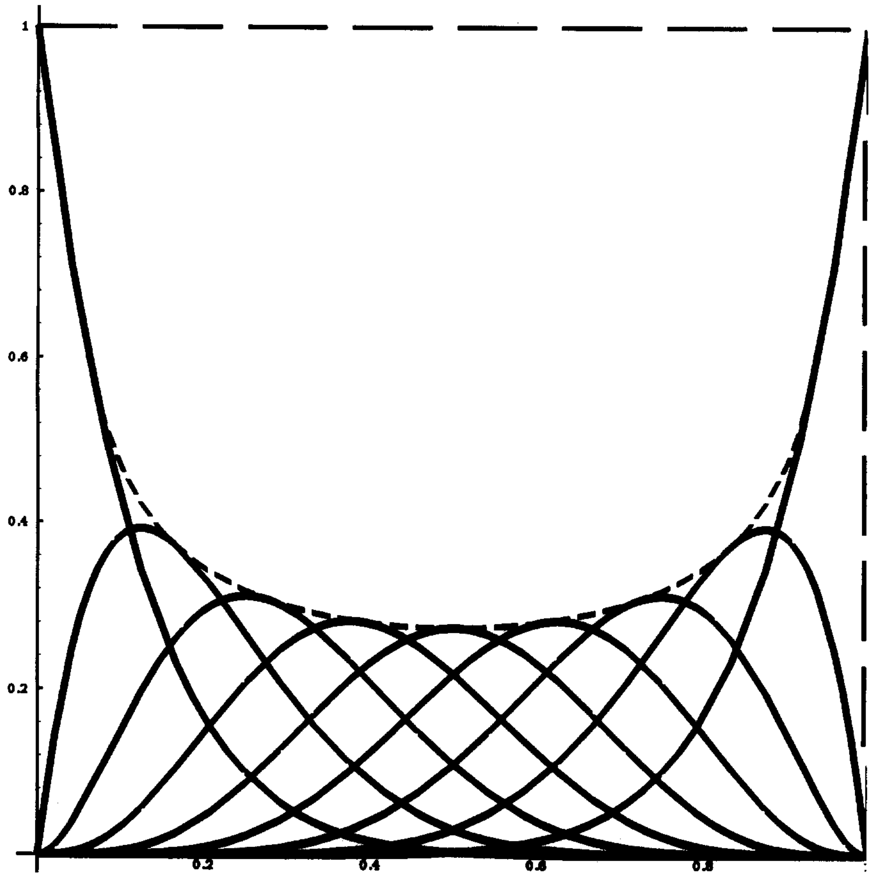


Figure 10.

We must find the solutions of

$$\frac{\partial}{\partial \lambda} G(t, y, \lambda) = \frac{\partial}{\partial \lambda} B_\lambda^n(t) = 0 \qquad \text{for} \quad (t, \lambda) \in (0,1) \times (0, n), \quad (4.3)$$

$$\frac{\partial}{\partial \lambda} B_\lambda^n(t) = t^\lambda (1 - t)^{n-\lambda} \left[\frac{\partial}{\partial \lambda} \binom{n}{\lambda} + \binom{n}{\lambda} \log \frac{t}{1-t} \right] = 0$$

$$\begin{aligned} \Leftrightarrow \quad \log \frac{t}{1-t} &= -\frac{\partial}{\partial \lambda} \log \binom{n}{\lambda} \\ &= \frac{\partial}{\partial \lambda} [\log \Gamma(\lambda + 1) + \log \Gamma(n - \lambda + 1)] \\ &= \psi(\lambda + 1) - \psi(n - \lambda + 1), \end{aligned} \qquad (4.4)$$

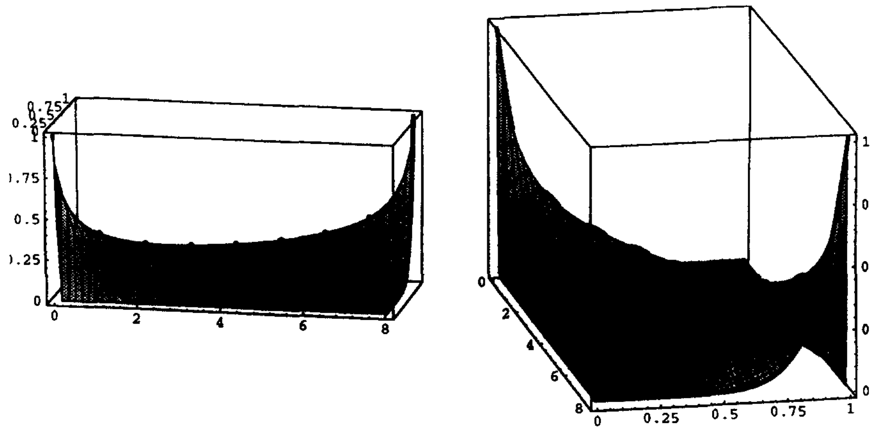


Figure 11.

where $\psi(x)$ is the Gauss psi function or polygamma function [1], which is defined as the derivative of $\log \Gamma(x)$ and can be expressed as

$$\psi(x+1) = -C + \sum_{k=1}^{\infty} \frac{x}{k(k+x)} \tag{4.5}$$

with $C \simeq 0.577216$, the Euler constant.

It is known [1, 9] that since $\log \Gamma(x)$ is convex for $x > 0$, then $\log \Gamma(\lambda + 1) + \log \Gamma(n - \lambda + 1)$ is convex too for $\lambda \in (0, n)$, and the function $l^n(\lambda)$ defined from (4.4) as

$$l^n(\lambda) := \psi(\lambda + 1) - \psi(n - \lambda + 1) \tag{4.6}$$

is a strictly monotone increasing function for all $\lambda \in (0, n)$, because it is the first derivative of a convex function. Solving (4.4) for t, we can define

$$t_\lambda := \tau_n(\lambda) = \frac{e^{l^n(\lambda)}}{1 + e^{l^n(\lambda)}}. \tag{4.7}$$

Since $e^x/(1 + e^x)$ is a strictly monotone increasing function as well as $l^n(\lambda)$, and since

$$\lim_{x \to -\infty} \frac{e^x}{1+e^x} = 0, \qquad \lim_{x \to +\infty} \frac{e^x}{1+e^x} = 1, \tag{4.8}$$

then $\tau_n(\lambda)$ is a strictly monotone increasing function too and is a bijective map from $(0, n)$ onto some subset of $(0, 1)$. Therefore, for each λ there exists a unique solution t_λ of (4.4), and $t_\lambda < t_{\lambda'}$ if $\lambda < \lambda'$. On the other hand, due to the convexity of $\log \Gamma$, it can be shown that $(\partial^2/\partial\lambda^2)B_\lambda^n(t)|_{t=\tau_n(\lambda)} < 0\ \forall\lambda \in (0, n)$. Therefore, (λ, t_λ) is a relative maximum with respect to λ.

Furthermore, due to the monotonicity of τ_n and the uniqueness of solutions of (4.4) for λ along the open interval $(0, n)$, the relative maximum is an absolute maximum too.

COROLLARY 4.5 *For every $i \in \{1, 2, \dots, n-1\}$ there exists a $t_i \in (0, 1)$ such that $B_i^n(t_i) > B_j^n(t_i)$ for $j \neq i$ and $t_i < t_j$ if $i < j$.*

Proof By Proposition 4.4, we only have to restrict the values of λ to the integers i. ∎

COROLLARY 4.6 *There exists a function k fulfilling the required conditions. This function is not unique.*

Proof From Remark 4.3 and Corollary 4.5 we have deduced the existence of a set of ordered nodes $\{t_i\}$ as required in Proposition 4.2; therefore we can construct the function $k_0(x)$ defined there. Any other function $k \geq k_0$ with $k(t_i) = k_0(t_i)$ is also a solution. ∎

Moreover, the freedom to choose solutions is wider: when we come back from $\lambda \in [0, n]$ to the discrete case, $i \in \{0, 1, \dots, n\}$, for every i, each point t_i that is a solution of (4.4) has an open interval of values t for which $B_i^n(t)$ is still greater than $B_j^n(t)$ for $j \neq i$, as we can see in Figure 10. We could change our choice of every t_i to another in its interval and begin the construction of another function k according to this new set of nodes, with the same freedom as before. It is worth noticing that every choice of nodes $\{t_i\}$ and every function k offers a different set of fuzzy numbers $\{\mu_i\}$ with different values and properties with *the same* Bézier curve as output.

Now that the existence of solutions has been proved, let us construct one of them. Although the function k_0 would be sufficient, it leads to a C^0 set of fuzzy numbers, therefore we have preferred a function k which takes account of the properties of the Bernstein polynomials and guarantees the smoothness of the fuzzy numbers. It is inspired by the envelope of the family $\{B_\lambda^n(x)\}$.

As we have seen in (4.7) and (4.8), $\tau_n(\lambda)$ maps $(0, n)$ onto a subset of $(0, 1)$. Let us determine it.

DEFINITION 4.7 *From* (4.7) *we define*

$$\tilde{t}_0 := \inf\{\tau_n(\lambda) \mid \lambda \in (0, n)\},$$
$$\tilde{t}_n := \sup\{\tau_n(\lambda) \mid \lambda \in (0, n)\}.$$

PROPOSITION 4.8 *If* $0 < n < +\infty$, *then* $0 < \tilde{t}_0 < \frac{1}{2} < \tilde{t}_n < 1$.

Proof From (4.4) and (4.6) plus continuity,

$$\begin{aligned} l^n(0) &= \lim_{\lambda\to 0} l^n(\lambda) = \lim_{\lambda \to 0} [\psi(\lambda + 1) - \psi(n - \lambda + 1)] \\ &= \psi(1) - \psi(n+1) \\ &= -C - \psi(n+1) = -\sum_{k=1}^{\infty} \frac{n}{k(k+n)} < 0 \qquad \forall n > 0. \end{aligned}$$

On the other hand, from the convexity of $n/x(x+n)$, we have

$$\begin{aligned} \sum_{k=1}^{\infty} \frac{n}{k(k+n)} &= \frac{n}{1+n} + \sum_{k=2}^{\infty} \frac{n}{k(k+n)} < \frac{n}{1+n} + \int_1^{\infty} \frac{n}{x(x+n)}\,dx \\ &= \frac{n}{1+n} - \log\frac{1}{1+n} < +\infty \qquad \text{if} \quad n < +\infty \end{aligned}$$

Therefore, for $0 < n < +\infty$,

$$-\infty < l^n(0) < 0.$$

Similarly,

$$l^n(n) = \psi(n+1) - \psi(1) = -l^n(0)$$

and $0 < l^n < +\infty$. Now, using (4.7) and the monotonicity of τ_n, we obtain

$$\tilde{t}_0 = \tau_n(0) = \frac{e^{-C-\psi(n+1)}}{1 + e^{-C-\psi(n+1)}},$$

$$\tilde{t}_n = \tau_n(n) = \frac{e^{C+\psi(n+1)}}{1 + e^{C+\psi(n+1)}}$$

and, with (4.8),

$$0 < \tilde{t}_0 < \tfrac{1}{2} < \tilde{t}_n < 1. \qquad \blacksquare$$

REMARK 4.9 There is no solution of (4.3) for $t \in [0, \tilde{t}_0)$, and therefore, $B_0^n(t)$ remains greater than $B_\lambda^n(t)$ in this interval. In a similar way, $B_n^n(t)$ remains greater than $B_\lambda^n(t)$ for $t \in (\tilde{t}_n, 1]$.

Since $\tau_n(\lambda)$ is a bijection from $(0, n)$ onto $(\tilde{t}_0, \tilde{t}_n)$, it has inverse in this interval, which will be denoted by $\lambda(t)$. Notice that, for every $t \in (\tilde{t}_0, \tilde{t}_n)$, $B_{\lambda(t)}^n(t)$ is an absolute maximum of $B_\lambda^n(t)$ with respect to λ, according to Proposition 4.4.

Collecting all these facts, we can now construct the nodes $\{t_i\}$, the function $k(t)$, and the fuzzy numbers $\{\mu_i\}$:

THEOREM 4.10 *Bézier curves are solutions of type* (2.3) *for a suitable choice of the nodes* $\{t_i\}$ *and the fuzzy numbers* $\{\mu_i\}$.

Proof

1. Take

$$t_0 = 0,$$
$$t_i = \tau_n(i), \qquad i = 1, \ldots, n-1,$$
$$t_n = 1.$$

2. Define $k : [0,1] \to \mathbb{R}^+ \setminus \{0\}$ as

$$k(t) = \begin{cases} B_0^n(t) & \text{if } 0 \le t \le \tilde{t}_0, \\ B_{\lambda(t)}^n(t) & \text{if } \tilde{t}_0 < t < \tilde{t}_n, \\ B_n^n(t) & \text{if } \tilde{t}_n \le t \le 1. \end{cases}$$

3. Define $\mu_i(t) = B_i^n(t)/k(t)$. This choice verifies

$$t_i < t_j \ \text{ if } i < j, \qquad \mu_i(t) \in [0,1], \qquad \mu_i(t_i) = 1$$

and $F_i(t) = \mu_i(t)/\sum_{j=0}^{n} \mu_j(t) = B_i^n(t)$, as needed. ∎

In Figure 12 the set of $\{\mu_i\}$ for $n = 8$ is represented.

About Bézier surfaces, let us recall that, given $(n+1) \times (m+1)$ control points $\vec{P}_{ij} \in \mathbb{R}^3$, the Bézier surface $\vec{B}(s,t)$ controlled by them is obtained, in its Bernstein form, by means of the formula

$$\vec{B}(s,t) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_i^n(s) B_j^m(t) \cdot \vec{P}_{ij} \qquad \text{for} \quad [s,t] \in [0,1] \times [0,1],$$

where the B_k^n are Bernstein's polynomials, as in (4.1) (see [3–5]). These surfaces are of the form (2.5) with $F_{ij} = B_i^n B_j^m$.

Figure 13 show a typical Bézier surface with its control net.

THEOREM 4.11 *Bézier surfaces are solutions of type* (2.5) *for a suitable choice of the nodes* $\{(s_i, t_j)\}$ *and the fuzzy sets* $\{\alpha_{ij}\}$.

Proof According to Theorem 4.10, we construct the nodes $\{s_i\}$ and $\{t_j\}$, the functions k_s and k_t and the fuzzy numbers $\{\mu_{s_i}\}$ and $\{\mu_{t_j}\}$ independently for s and t, and then we match the nodes and define the fuzzy sets α_{ij}, using the t-norm product, as $\alpha_{ij}(s,t) = \mu_{s_i}(s)\mu_{t_j}(t)$.

From here, Theorem 4.11 immediately follows. ∎

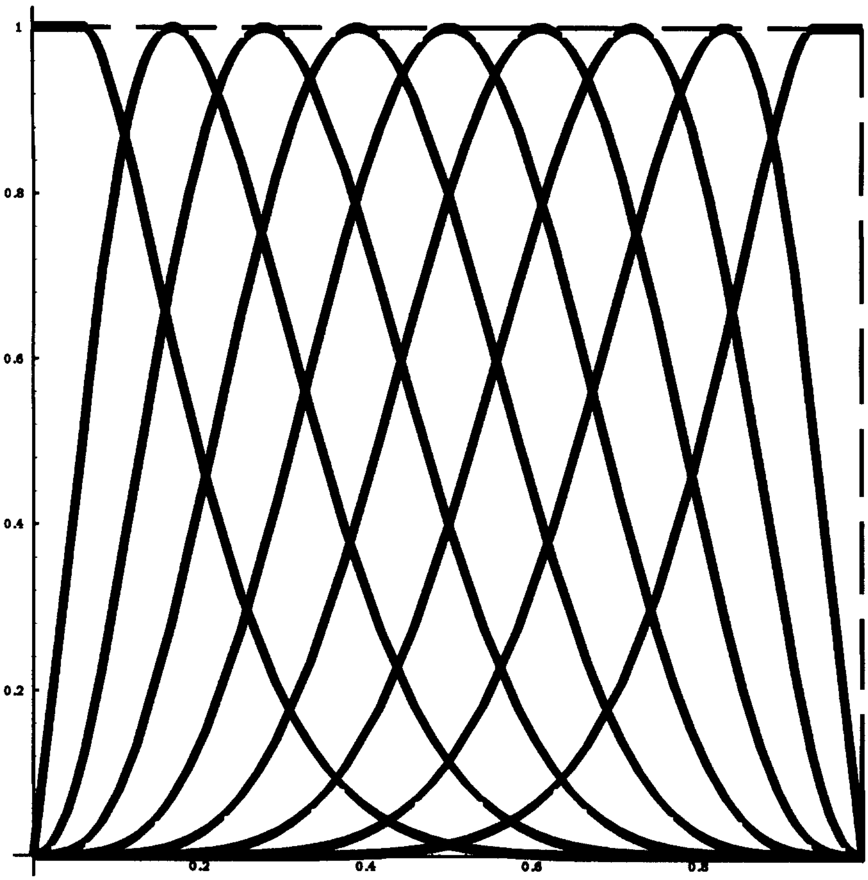


Figure 12.

5. SUMMARY

We have presented a model to design curves and surfaces in CAGD using fuzzy approximate reasoning. Apart from its practical use in concrete applications, it provides a new interpretation of the meaning of adjusting curves and surfaces to points and a new semantic in CAGD from the point of view of approximate reasoning and the fuzzy set theory. In addition, we have shown that some of the most popular methods, using the Bézier curves and surfaces, can be interpreted as members of our family.

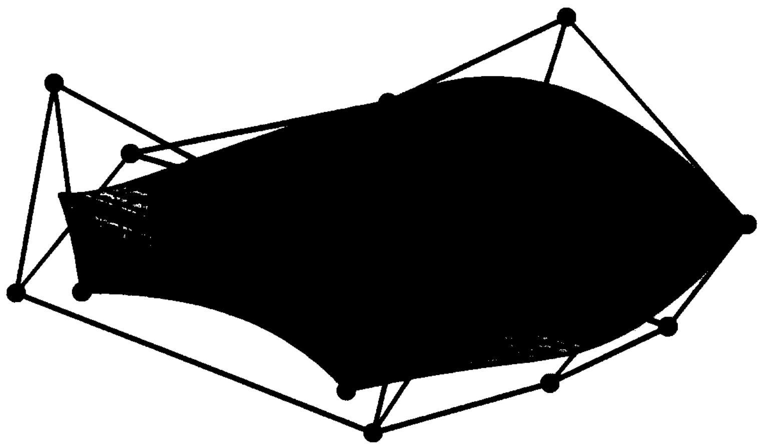


Figure 13.

References

1. Artin, E., *The Gamma Function*, Holt, Rinehart & Winston, New York, 1964.
2. Castro, J. L., Fuzzy logic controllers are universal approximators, *IEEE Trans. Sytem Man and Cybernetics* 25(4), 629–635, 1995.
3. Farin, G., *Curves and Surfaces for CAGD*, Academic, 1990.
4. Faux, I. D., and Pratt, M. J., *Computational Geometry for Design and Manufacture*, Ellis Horwood, 1985.
5. Hearn, D., and Baker, M. P., *Computer Graphics*, Prentice-Hall, 1994.
6. Jacas, J., Recasens, J., Fuzzy numbers and equality relations, *FUZZ'IEEE-93*, San Francisco, 1298–1301, 1993.
7. Kruse, R., Gebhard, J., and Klawoon, F., *Foundations of Fuzzy Systems*, Wiley, New York, 1994.
8. Piegl, L., On NURBS: A survey, *IEEE Trans. Comput. Graphics Appl.* 11(1), 55–71, 1991.
9. Rudin, W., *Principles of Mathematical Analysis*, 3rd ed., McGraw-Hill, 1976.
10. Takagi, H., and Sugeno, M., Fuzzy identification of systems and its applications to modelling and control, *IEEE Trans. Systems Man Cybernet.* 15, 116–132, 1985.
11. Wang, L.-X., Fuzzy systems are universal approximators, *FUZZ'IEEE-92*, San Diego, 1163–1170, 1992.